Fediz Configuration

FEDIZ PLUGIN CONFIGURATION

This page describes the Fediz configuration file referenced by the security interceptor of the Servlet Container (eg. authenticator in Tomcat/Jetty).

The Fediz configuration information is used to publish the federation Metadata document which is described here

Example

The following example shows the minimum configuration for Fediz.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<FedizConfig>
    <contextConfig name="/fedizhelloworld">
        <audienceUris>

<audienceItem>https://localhost:8443/fedizhelloworld</audienceIte
m>
        </audienceUris>
        <certificateStores>
            <trustManager>
                <keyStore file="conf/stsstore.jks"
password="stsspass" type="JKS" />
            </trustManager>
        </certificateStores>
        <trustedIssuers>
            <issuer certificateValidation="PeerTrust" />
        </trustedIssuers>
        <protocol
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="federationProtocolType" version="1.2">

<issuer>https://localhost:9443/fediz-idp/federation/</issuer>
        </protocol>
    </contextConfig>
</FedizConfig>
```

The protocol element declares that the WS-Federation protocol is being used. The issuer element shows the URL to which authenticated requests will be redirected with a SignIn request.

The IDP issues a SAML token which must be validated by the plugin. The validation requires the certificate store of the Certificate Authority(ies) of the certificate which signed the SAML token. This is defined in `certificateStore`. The signing certificate itself is not required because `certificateValidation` is set to `ChainTrust`. The `subject` defines the trusted signing certificate using the subject as a regular expression.
Finally, the audience URI is validated against the audience restriction in the SAML token.

Configuration reference

XML element	Name	Use	Description
audienceUris	Audience URI	Required	The values of the list of audie are verified against the eleme `nceRestriction` in the SAML
certificateStores	Trusted certificate store	Required	The list of keystores (JKS, PE includes at least the certificate Certificate Authorities (CA) wh signed the certificate which is sign the SAML token. If the file location is not fully q needs to be relative to the Co home directory
trustedIssuers	Trusted Issuers	Required	There are two ways to configu trusted issuer (IDP). Either yo configure the subject name ar CA(s) who signed the certifica IDP (`certificateValidatio Trust`) or you configure the ce of the IDP and the CA(s) who (`certificateValidation=Pe`)
maximumClockSkew	Maximum Clock Skew	Optional	Maximum allowable time diffe between the system clocks of and RP. Default 5 seconds.
tokenReplayCache	Token Replay Cache	Optional	The ReplayCache implementa use to cache tokens. The defa implementation based on EH(
signingKey	Key for Signature	Optional	If configured, the published (WS-Federation) Metadata do s signed by this key. Otherwis signed.
tokenDecryptionKey	Decryption Key	Optional	A Keystore used to decrypt ar encrypted token.

tokenExpirationValidation	Token Expiration Validation	Optional	Decision whether the token va (e.g. lifetime) shall be perform every request (true) or only or initial authentication (false). Th default is "false".

WS-Federation protocol configuration reference

XML element	Name	Use	Metadata
issuer	Issuer URL	Required	PassiveRequestorEndpoint
realm	Realm	Optional	TargetScope
authenticationType	Authentication Type	Optional	NA

roleURI	Role Claim URI	Optional	NA
roleDelimiter	Role Value Delimiter	Optional	NA
claimTypesRequested	Requested claims	Optional	ClaimTypesRequested

homeRealm	Home Realm	Optional	NA
freshness	Freshness	Optional	NA
request	Request	Optional	NA
tokenValidators	TokenValidators	Optional	NA

Attributes resolved at runtime

The following attributes can be either configured statically at deployment time or dynamically when the initial request is received:

- authenticationType
- homeRealm
- issuer
- realm

These configuration elements allows for configuring a CallbackHandler which gets a Callback object where the appropriate value must be set. The CallbackHandler implementation has access to the HttpServletRequest. The XML attribute `type` must be set to `Class`.

For more information see Fediz Extensions.

Advanced example

The following example defines the required claims and configures a custom callback handler to define some configuration values at runtime.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<FedizConfig>
    <contextConfig name="/fedizhelloworld">
        <audienceUris>

<audienceItem>https://localhost:8443/fedizhelloworld</audienceIte
m>
        </audienceUris>
        <certificateStores>
            <keyStore file="conf/stsstore.jks"
password="stsspass" type="JKS" />
        </certificateStores>
        <maximumClockSkew>10</maximumClockSkew>
        <trustedIssuers>
            <issuer certificateValidation="PeerTrust" />
        </trustedIssuers>
        <signingKey keyPassword="tompass">
            <keyStore file="tomcatKeystore.jks"
password="tompass" type="JKS" />
        </signingKey>
        <protocol
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="federationProtocolType" version="1.2">

<issuer>https://localhost:9443/fediz-idp/federation/</issuer>
            <roleDelimiter>,</roleDelimiter>

<roleURI>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/ro
```

```
le</roleURI>
            <claimTypesRequested>
                <claimType
type="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/role"
optional="true" />
            </claimTypesRequested>
            <authenticationType type="String"
value="http://docs.oasis-open.org/wsfed/authorization/200706/auth
ntypes/smartcard" />
            <homeRealm type="Class"
value="example.HomeRealmCallbackHandler" />
            <tokenValidators>

<validator>org.apache.cxf.fediz.core.CustomValidator</validator>
            </tokenValidators>
        </protocol>
```

```
    </contextConfig>
</FedizConfig>
```